

## PROYECTO 1 ENTREGA 1

### APLICACIONES WEB ESCALABLES EN UN ENTORNO TRADICIONAL

#### OBJETIVOS

- Desarrollar una aplicación Web que involucre el uso de servidores Web y procesos en background.
- Identificar arquitecturas y patrones de referencia para aplicaciones Web ejecutadas en un entorno tradicional (con servidores e infraestructura propia), que involucren el uso de servidores Web, servidores de bases de datos y procesos en background.

#### DESCRIPCIÓN DE LA APLICACIÓN WEB A DESARROLLAR

Una nueva compañía de cloud denominada SmartTools (ver ejemplo real en <https://www.socialtools.me/>) desea crear una aplicación de Software como Servicio (SaaS) que será ofrecida a empresas de todos los tamaños (pequeñas, medianas y grandes) para que los clientes de esas empresas puedan realizar fácilmente concursos de videos con su base de clientes.

El modelo general de funcionamiento del SaaS se basa en que un administrador de una empresa entra al portal [www.smarttools.com](http://www.smarttools.com) y crea una cuenta. Una vez la cuenta ha sido creada el administrador puede proceder a configurar los concursos de video que esa empresa desea ofrecer a sus clientes. **Por cada concurso publicado el sistema le genera una URL única** (que puede ser cambiada por el administrador), la cual el administrador puede publicar en TV, su portal Web o enviar a sus clientes a través de email para que estos puedan entrar y subir los videos a cada concurso, los cuales deben estar relacionados con los productos vendidos u ofrecidos por la empresa. Los videos serán revisados por el equipo de marketing de la empresa y el mejor video de cada concurso ganará un fabuloso premio.

**Para esto la compañía de cloud debe desarrollar una aplicación Web que les permita a las empresas interesadas en utilizar la plataforma:**

1. **(5%)** Posibilitar que un administrador de la empresa (que es el público objetivo que se espera visite el sitio Web) que visite el home del sitio pueda ver la información básica del SaaS, conozca los valores agregados del servicio y pueda proceder a crear una cuenta en el sitio Web. Para crear una cuenta el administrador debe ingresar: sus nombres, sus apellidos, el email, la contraseña y la confirmación de la contraseña.
2. **(15%)** Permitir que una vez iniciada sesión el administrador pueda proceder a configurar los concursos. De cada concurso el administrador debe configurar el nombre del concurso, el banner o imagen del concurso, la URL única a todo el sistema (a través de la cual van a poder acceder los usuarios finales), la fecha de inicio del concurso, la fecha de fin del concurso y la descripción del premio que ganará el mejor video. El administrador debe poder gestionar (realizar el CRUD – Create/Read/Update/Delete) los concursos.
3. **(10%)** Permitir que vez inicie sesión el administrador de una empresa, éste pueda ver el detalle de un concurso, el cual debe incluir el listado de todos los videos que han sido subidos por los usuarios al concurso. El listado de videos de cada concurso debe estar ordenado de forma descendente por fecha de creación y se debe utilizar paginación de a 50 videos. En el listado el administrador podrá ver para cada video subido el email del usuario que lo subió, sus nombres y apellidos, la fecha en que se subió el video, el estado del video (“En proceso” o “Convertido”), y puede previsualizar el video original y el convertido (en caso de que el estado sea “Convertido”).

**Por cada concurso creado la compañía de cloud debe proveer una página Web que los clientes de las empresas visitarán y podrán:**

4. **(5%)** Ver en la página principal de cada concurso (Home del Concurso) todos los videos subidos por los usuarios ordenados del video más reciente al más antiguo. En caso de existir muchos videos se recomienda hacer paginación si hay más de 50 videos publicados.
5. **(8%)** Reproducir los videos cargados por los usuarios en el home, para esto se recomienda usar la herramienta JW Player (<http://www.longtailvideo.com/>). Para ver un video no es necesario que el usuario inicie sesión en el sitio Web.
6. **(12%)** Desde el Home del Concurso, el usuario debe poder subir un nuevo video, para ello el usuario debe ingresar: los nombres, los apellidos, el email, el video y un mensaje asociado al video. El usuario puede subir un video (o varios realizando envíos individuales) en cualquier formato (ej. AVI,WMV, FLV, MP4, etc.). El usuario envía el video y en ese

momento el video queda en estado “En proceso” y aún NO deberá aparecer el video en el listado del Home del Concurso. El video original quedará almacenado en un sistema de archivos.

Con el fin de lograr que los videos los puedan ver los usuarios del sitio Web desde cualquier dispositivo, es necesario que todo video subido por un usuario sea convertido a formato MP4 usando el códec de video H.264 y el códec de audio AAC. Para evitar que el usuario se demore esperando que los videos sean convertidos, el proceso de conversión se hará a través de tareas en background o batch. Una vez el video haya sido convertido a este formato (utilizando la herramienta FFmpeg <http://www.ffmpeg.org/>) el estado del video cambiará su estado a “Convertido”. A partir de ese momento el video deberá aparecer en el listado del Home del Concurso.

Para cada video el usuario deberá ingresar obligatoriamente un mensaje de porqué le gusta el producto relacionado en el video. Tan pronto el usuario presiona la opción de “Enviar Video” el sistema le muestra un mensaje del estilo “Hemos recibido tu video y los estamos procesado para que sea publicado. Tan pronto el video quede publicado en la página del concurso te notificaremos por email.”.

**Por la conversión de videos la compañía de cloud deberá ejecutar un proceso en batch.**

7. **(20%)** El sistema deberá ejecutar un proceso periódicamente (cron a nivel de sistema operativo o de servidor de aplicaciones) que se encargue de consultar en la base de datos todos los videos que están en el estado “En proceso” y para cada video en ese estado el sistema deberá:
  - Convertir el video al formato MP4 usando el códec de video H.264 y el códec de audio AAC utilizando la herramienta FFmpeg <http://www.ffmpeg.org/>).
  - Guardar el video convertido (sin borrar el video original) en un sistema de archivos y cambiar el estado del video a “Generado”.
  - Enviar al usuario final un email informándole que el video ya ha sido publicado en la página pública del concurso.

Todas las anteriores funcionalidades deberán ofrecerse desde una sola aplicación web y un proceso en batch o cron que será desarrollado por la compañía de cloud. Dado que esta es la primera iteración del producto otras funcionalidades relacionadas con precios, facturación, evaluación de videos, publicación de ganadores, entre otros, no serán incluidas en esta fase de desarrollo. El diseño de las interfaces de usuario es de libre elección, sin embargo, por facilidad se recomienda utilizar frameworks de front-end como Bootstrap (<http://getbootstrap.com/>) o Foundation (<http://foundation.zurb.com/>).

## RECOMENDACIONES Y CONSIDERACIONES

En este proyecto se deberá desarrollar la aplicación Web con las funcionalidades mencionadas. La aplicación Web deberá ser desarrollada en los lenguajes de programación Java, Python, Ruby, Scala o Javascript, y los frameworks de desarrollo con los que esté más familiarizado. **Por temas de agilidad en el desarrollo de este proyecto y para los demás proyectos del curso se recomienda el uso de frameworks de desarrollo ágil como “Ruby on Rails”** (<http://rubyonrails.org/>), **“Django”** (<https://www.djangoproject.com/>), **“Play”** (<https://www.playframework.com/>) o **“NodeJS/Express”** (<https://nodejs.org>). La aplicación deberá tener como mínimo dos componentes para su ejecución en un entorno local (en servidores propios): el servidor web y el servidor de base de datos. Podrán utilizar los servidores de aplicaciones, de bases de datos relacionales y las herramientas de desarrollo que más se le faciliten. El servidor web utilizado debe estar en capacidad de manejar solicitudes de forma concurrente.

Recomendamos que utilice el framework con el que esté más familiarizado, si no está familiarizado con ninguno de los anteriores frameworks de desarrollo ágil, **le recomendamos desarrollar la aplicación en Ruby on Rails dada la agilidad para desarrollar la aplicación y su facilidad para el despliegue escalable de la aplicación sobre servicios de proveedores IaaS como Amazon Web Services y proveedores de PaaS como Heroku** (lo cual se realizará en las entregas 2, 3 y 4 del proyecto 1).

Se debe tener en cuenta que el sitio Web debe estar diseñado para que decenas de clientes puedan subir videos de manera concurrente, por lo cual es importante que la conversión de videos se haga en batch, es decir, que una vez un usuario sube un video se le muestra una interfaz donde se le indica que el video será convertido. Luego debe haber procesos (crons) en background que estén encargados de convertir los videos, guardar los videos convertidos, cambiar el estado de los videos a “Generado” y enviar un email al usuario indicándole que el video ya ha sido publicado.

**Nota Importante:** Para la socialización de la entrega deben tener disponibles videos de prueba en formato mpg, mp4, avi, flv que permitan mostrar las funcionalidades de la aplicación.

## ENTREGABLES ADICIONALES

1. **(8%) Escenario 1.** Se deberá definir un escenario donde se pueda probar cuál es la máxima cantidad de peticiones HTTP por minuto que soporta la página de un concurso con 30 videos publicados. Para hacer pruebas de carga se sugiere utilizar la herramienta JMeter que podrá instalar en un entorno local. Las pruebas de stress deberán realizarlas desde otros equipos diferentes a los utilizados para ejecutar el servidor web y el servidor de base de datos. El escenario y los resultados de las pruebas de estrés deberán ser documentados

con gráficas que ilustren cómo se comporta el sistema a medida que el número de clientes accediendo a la página de un concurso se incrementa hasta llegar al punto de degradar completamente el rendimiento del sistema.

**Restricciones del escenario.** En las pruebas de stress el tiempo de respuesta promedio de la página debe ser de máximo 1.500 ms, si este tiempo no se cumple, se concluye que el sistema NO soporta la cantidad de peticiones de la prueba. En caso de que durante una prueba se generen más de un 1% de errores en las peticiones de la prueba, se concluye que el sistema NO soporta la cantidad de peticiones de la prueba.

**El nombre de este documento deberá ser: “Proyecto 1 entrega 1 - Escenario y Pruebas de Stress Aplicación Web.pdf”.**

2. **(7%) Escenario 2.** Se deberá definir un escenario donde se pueda probar cuál es la máxima cantidad de videos que pueden ser procesados por minuto en la aplicación local. Para hacer pruebas de carga se sugiere utilizar la herramienta JMeter que podrá instalar en un entorno local. Las pruebas de stress deberán realizarlas desde otros equipos diferentes a los utilizados para ejecutar el servidor web y el servidor de base de datos. El escenario y los resultados de las pruebas de estrés deberán ser documentados con gráficas que ilustren cómo se comporta el sistema a medida que el número de clientes convirtiendo videos se incrementa, hasta llegar al punto en que el tiempo para iniciar la conversión de un archivo de video enviado por un usuario supere media hora (1800 segundos).

**Restricciones del escenario.** El video enviado a convertir durante las pruebas debe ser en formato AVI de mínimo 120 segundos de video.

**El nombre de este documento deberá ser: “Proyecto 1 entrega 1- Escenario y Pruebas de Stress Aplicación Batch.pdf”.**

3. **(10%)** Se deberá entregar un documento donde se describa la arquitectura de la aplicación, las conclusiones identificadas con las pruebas de stress ejecutadas y las consideraciones que deben ser tenidas en cuenta para que la aplicación pueda escalar a cientos de administradores y usuarios finales de los concursos que van a estar utilizando el sitio [www.smarttools.com](http://www.smarttools.com) de manera concurrente. En este documento se deben describir las limitaciones del desarrollo realizado y la infraestructura utilizada con respecto a la escalabilidad de la solución desarrollada.

**El nombre de este documento deberá ser: “Proyecto 1 entrega 1 - Arquitectura, conclusiones y consideraciones.pdf”**

**Nota Importante:** Las pruebas de stress de los escenarios 1 y 2 deben ejecutarse de forma aislada.

## EQUIPO DE TRABAJO

El proyecto deberá ser realizado en grupos de mínimo 2 y máximo 3 estudiantes.

## ESQUEMA DE EVALUACIÓN

La distribución de la calificación del proyecto está distribuida de la siguiente manera:

- Verificación funcional de los requerimientos de la aplicación: **75%**
- Documento de escenarios y resultados de las pruebas de stress: **15%**
- Documento de la arquitectura y consideraciones de la aplicación: **10%**

## LUGAR Y FORMATO DE ENTREGA

La entrega deberá ser realizada de la siguiente manera:

- Crear un release del código fuente en el repositorio del grupo en GitHub.
- Entregar todos los documentos solicitados en formato PDF, en el espacio de Moodle.

## FECHA DE ENTREGA

La entrega 1 del proyecto deberá ser realizada de acuerdo a las fechas publicadas en la plataforma Moodle.

- La socialización del proyecto estará a cargo de uno de los miembros del equipo de trabajo seleccionado aleatoriamente por ello es importante que los otros miembros del equipo conozcan en detalle el funcionamiento de toda la aplicación. Para el día de la sustentación el sitio Web debe estar funcionando ya sea en los equipos de la universidad o en sus máquinas personales.